Use Case 3 - Comment and Rate Web Services

Purpose

A registry can soon become of little value if it is cluttered with numerous defunct or low quality services. If 25% of the numbers in the phone book were wrong you would be inclined to give up using the phone book. This is a problem for relatively small, manageable internal registries and that problem is magnified exponentially with a widely distributed, global higher education registry. There needs to be some method of rating web services so that potential consumers of those services can separate the wheat from the chaff.

There are at least two or three different approaches that could be taken in order to accomplish this goal. As we iterate over this use case, we will drive down to one optimal approach, or perhaps combine two or more compatible approaches. The bottom line is that we want to provide a level of confidence to participants in the EdUnify universe that it is possible to efficiently locate bona fide, 'certified', high quality services via the registry with minimal human overhead.

Actors

Application Administrators, System Administrators

Applications

All participating EdUnify applications.

High-level Story (Abstract)

A participating EdUnify entity (institution, vendor, government department) deploys a web service such as the Academic History Web Service and successfully publishes that service to the EdUnify distributed registry. Metadata is attached to that registry entry to document the quality expectations for that service and whether those expectations are being met. That metadata may be a combination of information supplied by the publisher, consuming applications/monitoring tools and administrators. The metadata should be reliable at any point in time, reflecting the true status of the service. The information can be leveraged when performing search operations against the EdUnify distributed registry. Administrators and/or applications can review the metadata in order to determine whether a service may meet their needs or not.

Detailed Story (with illustrations and schematics as needed)

There are three classes of metadata that could potentially be attached to each service entry to meet our goal:

1. Static attributes that document the Service Level Agreement (SLA) that the publisher is committing to (e.g., availability, response time, adherence to industry standard).
2. Dynamic attributes that are continually updated based on calls to that service or monitoring of that service.
3. Ratings and comments entered manually by application/system administrators who have had experience using, or attempting to use the service (very similar to Amazon Customer Reviews with a rating of 1 to 5).

The SLA and Ratings/Comments information both rely on human data entry and so are prone to being unreliable. Particularly with the Ratings we must consider whether the community will take the time to enter the information. A hybrid approach may work quite well where we establish specific SLA attributes, automate the capture of metrics against those attributes and in addition, allow human administrators to rate and comment.

Having said all that, there are significant benefits to keeping things as simple as possible. For example, what if the only metric was the number of times that the service has been invoked. Services that do not work, or are of low quality/value will fall to the bottom of the heap for a given producer. Services that are of high value will be used.

To Do

This is just an initial draft. A discussion document more than anything. Some fundamental decisions are needed and then we can drill down into further details concerning exactly what metadata is needed and how it gets populated.